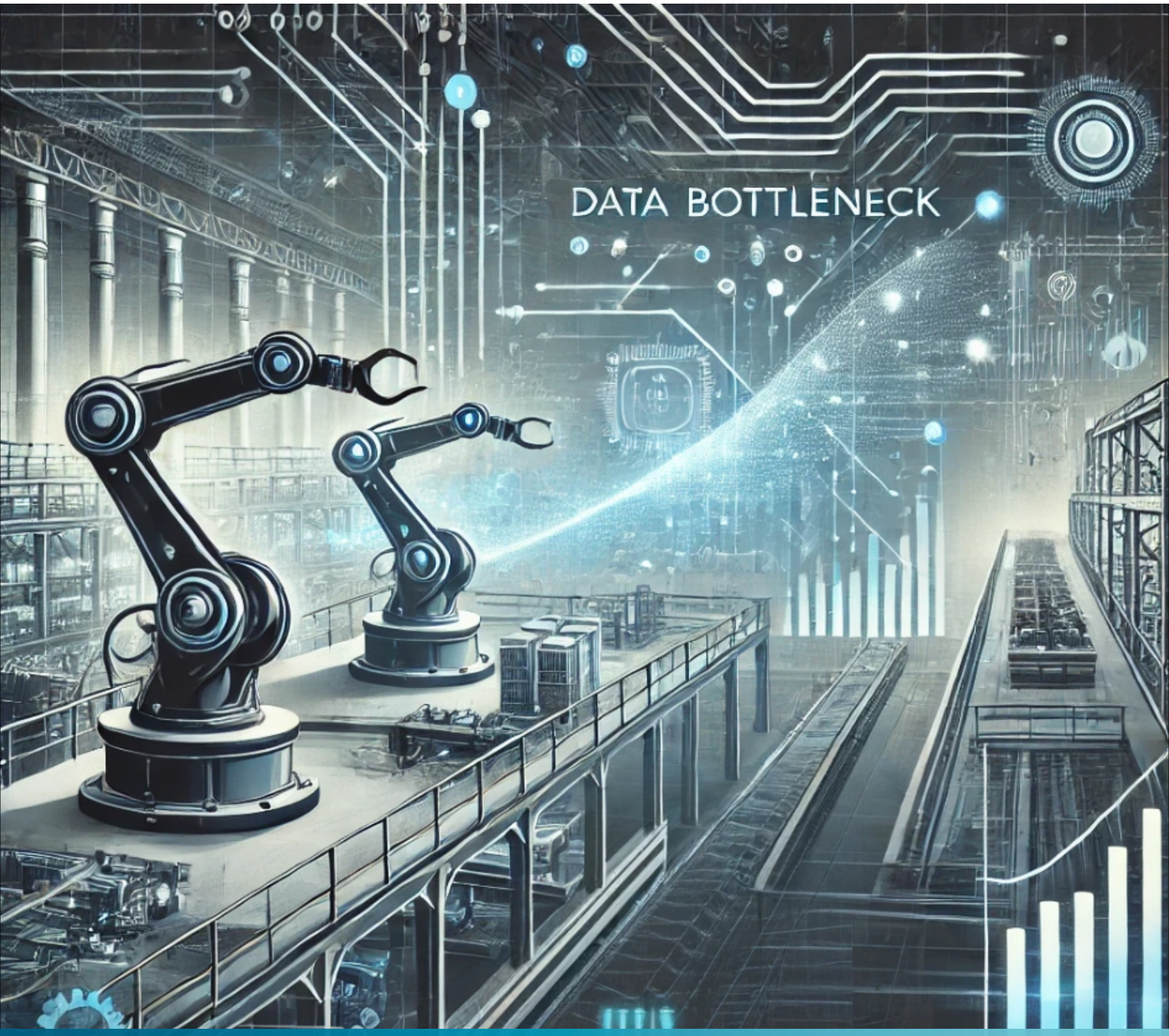
PROKI KARLSRUHE

KIT Karlsruhe Institute of Technology

Interaktive Echtzeitsysteme IES

wbk Institute of Production Science



Managing AI in Manufacturing Systems

Solving the Data Bottleneck

Contents

Abstract

The Data Bottleneck refers to the challenge of ensuring the availability of the right data at the right time in AI-driven projects. Early stages often involve uncertainty about when, how, and how much data will be required. The proposed approach focuses on estimating data requirements and determining when the data is needed at each phase of the AI lifecycle. This includes identifying critical data dependencies, ensuring data quality, managing imbalanced datasets, and implementing post-deployment monitoring to adapt to data shifts. By addressing these issues, organizations can enhance fairness, accuracy, and adaptability while sustaining model performance. Effective data bottleneck management empowers organizations to unify their data, improving trust, accessibility, and control. This approach supports key business objectives while enabling the development of reliable, scalable, and adaptable AI systems.

Funding

This Guideline is funded by the Federal Ministry of Education and Research (BMBF) within the "Demonstration and Transfer Network AI in Production (ProKI-Netz)" funding measures, under the program "The Future of Value Creation – Research in Production, Services, and Work," and is supervised by the Project Management Agency Karlsruhe (PTKA).

The authors are responsible for the content of this publication. Wherever both the feminine and masculine forms are not explicitly mentioned, the language is intended to address both women and men equally.

Main Author

Shahenda Youssef, MEng.
Karlsruhe Institute of Technology (KIT), Department of Informatics
Institute for Anthropomatics and Robotics (IAR)
Vision and Fusion Laboratory (IES), chaired by Prof. Dr.-Ing. Jürgen Beyerer *
Shahenda.youssef@kit.edu

Co-Authors

Dr.-Ing. Julius Pfrommer †
Fraunhofer Institute for Optronics, System Technologies and Image Exploitation IOSB
Julius.Pfrommer@iosb.fraunhofer.de

Florian Oexle, M.Sc.
Karlsruhe Institute of Technology (KIT)
wbk Institute of Production Science
Florian.oexle@kit.edu

Malte Hansjosten, M.Sc.
Karlsruhe Institute of Technology (KIT)
wbk Institute of Production Science
Malte.Hansjosten@kit.edu

* Vision and Fusion Laboratory: https://ies.iar.kit.edu/
† Acknowledged for his valuable contributions to this work, independent of project funding.

1. The Data Bottleneck for AI in Industry

The availability of the right data at the right time is a key challenge in AI-driven projects. At the start of a project, there is uncertainty about when, how, and how much data will be needed. Data acquisition is often a critical dependency in a project, requiring careful planning and coordination to ensure its timely availability. Understanding the data requirements in each phase of the AI project and ensuring high data quality are critical for developing effective and reliable AI systems.

AI in Industry

Artificial Intelligence (AI) in Industry involves the end-to-end process of designing, developing, deploying, and maintaining AI systems to improve operational efficiency, boost productivity, and enhance decision-making across manufacturing processes such as predictive maintenance, quality control, and robotics for automation. It optimizes supply chains, enhances customer service with chatbots, and supports energy management for sustainability. By enabling smarter, scalable, and adaptive operations, AI drives innovation and transformation across diverse industrial sectors.

These AI systems analyze vast amounts of data to uncover the reasons behind past events, predict future outcomes, and offer actionable insights to optimize performance. However, due to the intricacies associated with AI projects, there is a critical need for well-structured, robust methodologies to ensure successful outcomes. One such approach is the Cross-Industry Standard Process for Data Mining (CRISP-DM) [1], which offers a systematic method for planning, organizing, and executing data mining projects. The PAISE® model [2] emphasizes a systematic and standardized development process for AI-based system engineering. In Andrew Ng's machine learning lifecycle [3], data are emphasized as the foundation for model success. It is a data-centric approach, where high-quality, representative, and accurate data drive better performance rather than focusing on complex model designs. In each lifecycle stage, from scoping to deployment, ensuring data quality enhances the model's ability to generalize and adapt. Focusing on data issues often yields better improvements during error analysis than model tuning.

AI in industry presents significant challenges, especially when it comes to managing and utilizing data effectively. Inconsistent, incomplete, or biased data can lead to inaccurate predictions and suboptimal decisions. In addition, the complexity of manufacturing environments often requires handling large, dynamic datasets, which can strain data collection, processing, and storage capabilities. Nearly 85% of AI projects fail due to poor data quality. The effectiveness of AI models depends as much on the quality of data as on the algorithms themselves. Experts even estimate that data scientists spend 60-80% of their time on data preparation. Addressing these data-related issues is essential for reliable AI-driven insights and managing the time required for successful implementation.

Data Bottleneck

To manage data needs effectively, it is essential to consider the requirements at different stages of the project lifecycle: before the project starts, during the project, and after deployment. Each phase follows a consistent approach to ensure comprehensive problem management. The initial focus is on identifying potential issues, assessing their impact, and establishing strategies to address them proactively.

We coin the term **Data Bottleneck** to refer to the challenge of managing the availability of the right data at the right time. At the start of a project, there is uncertainty about when and how much data is needed. The acquisition of additional data is often in the critical path of a project and may depend on previous work like further process instrumentation. Understanding the data requirements in each phase of an AI project and ensuring high data quality are critical for developing effective and reliable AI systems.

This white paper will describe the major challenges and differences compared to traditional project management in manufacturing. We propose an integrated approach to identify the data bottleneck, propose effective solutions, and manage these challenges throughout the project lifecycle. Additionally, detailed manufacturing use case implementations will be discussed. The target audience for this guideline is project leaders and decision-makers. It provides them with a framework for project management, addressing key challenges, and facilitating effective interaction with technical development teams.

2. Identifying the Data Bottleneck

Each AI methodology offers unique advantages and considerations for successful integration into manufacturing processes. However, high-quality data is crucial for their success, presenting challenges across the AI lifecycle, from collection to deployment. These challenges influence scalability, performance, and project timelines, causing delays and uncertainties. Addressing these challenges proactively is essential for building reliable and efficient AI systems.

2.1. Data-Driven AI Method Selection

Define Project Goals

Before selecting an AI methodology, it is crucial to clearly define what you want to achieve with AI. This initial stage involves identifying your objectives and determining the specific problems you aim to solve. Recognizing the complexity and size of the AI projects ensures that the chosen AI tool can effectively meet the project's needs.

Not all problems are feasible or appropriate for AI solutions, making it essential to evaluate whether AI is the right tool for the task. Some challenges may require more foundational changes before AI can be effectively implemented.

Project leaders should consider the following questions

- What is the primary objective of the project?
- Which problems are suitable and feasible to solve using machine learning?
- What are the expected outcomes, and how will success be measured?
- What are the constraints, such as timelines, budgets, or available resources?
- How will the AI system integrate with existing processes or systems?
- What are the potential risks or ethical considerations, and how will they be managed?

By addressing these questions early on, project leaders can ensure that the goals are realistic, measurable, and aligned with both business priorities and technical capabilities, paving the way for a focused and efficient project execution. Furthermore, connecting the project goals with the specific characteristics of various AI methodologies enables teams to select the most suitable approach, ensuring that the solution is both technically feasible and optimized to effectively tackle the identified problem.

Data Considerations in AI Method Selection

The effectiveness of AI relies heavily on the quality, quantity, and diversity of the data provided. Assessing the available data significantly influences the choice of AI methodology for a manufacturing problem.

Understanding the types of data involved is crucial, it can be categorized into several types as shown in Table 1: structured data, unstructured data, semi-structured data, and time series data. Each serves a specific role in the model development and deployment phases, highlighting the importance of selecting the right methodology to effectively process and analyze the data.

The data's nature—labeled or unlabeled—plays a crucial role in determining the appropriate branch of Machine Learning (ML), a subset of AI that focuses on developing algorithms and models capable of learning patterns from data to make predictions or decisions without explicit programming, as illustrated in Figure 1.

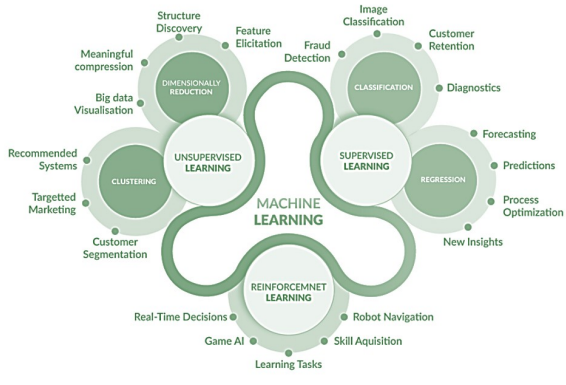


Figure 1: Machine Learning Branches.

Supervised Learning: This approach is used when the data is labeled, meaning each input has a corresponding output. For structured data, supervised learning is well-suited for tasks like classification and regression, where the model learns by mapping inputs to outputs through examples. Additionally, labeled time series data is often used in super-

Category	Description
Structured Data	Data organized in a tabular format with rows and columns, making it easy to understand and analyze. **Example** Spreadsheets, SQL databases, financial transactions, customer records, maintenance logs. **Use cases** Regression, classification, and clustering.
Unstructured Data	Data that does not have a predefined format, making it more challenging to process and analyze. **Example** Text documents, images, audio files, and videos. **Use cases** NLP for text analysis, computer vision for image and video recognition, and speech recognition systems.
Semi-Structured Data	Data that does not fit neatly into tables but still contains some organizational structure. **Example** JSON and XML files, email content, and sensor data with metadata. **Use cases** Web scraping results, hierarchical data analysis, and data exchange between systems.
Time Series Data	Data points collected or recorded at specific time intervals. **Example** Stock prices, temperature readings, and server logs. **Use cases** Forecasting, anomaly detection, and analyzing trends over time.

Table 1: The fundamental categories of data.

vised learning for forecasting or trend analysis.

Unsupervised Learning: When data lacks labels, unsupervised learning becomes essential. This approach uncovers hidden patterns, relationships, or structures in the data. For unstructured data, unsupervised methods are commonly used for clustering or anomaly detection. Similarly, semi-structured data, can benefit from unsupervised learning to discover underlying relationships or reduce dimensionality.

Reinforcement Learning: This method requires an environment where an AI agent interacts with its surroundings, receives feedback, and learns to optimize its actions over time. Time series data is often integrated into reinforcement learning scenarios, where sequential decisions and temporal feedback play a critical role. This makes reinforcement learning especially powerful for dynamic, decision-based environments such as robotics or automated systems.

Each ML branch serves specific purposes and depends heavily on the nature of the data and the problem being addressed, emphasizing the importance of aligning the choice of ML approach with the project's goals and available data.

AI methodology involves either Machine Learning (ML) or Deep Learning (DL), with the latter utilizing neural networks to accomplish its objectives. The choice between ML and DL, directly impacts the data requirements of an AI project, affecting aspects like the amount of data, preprocessing efforts, and data diversity as shown in Table 2.

For projects with limited data, smaller budgets, or simpler problems, ML is a better choice. For tasks involving unstructured data, complex patterns, or scalability, DL is more suitable—provided there is sufficient data volume, diversity, and quality. Balancing these factors ensures an optimal approach for the AI project's success.

2.2. Estimation of Data Requirements

Scaling Laws

How do model performance scales with respect to three key factors: model size, dataset size, and computational power?

In neural networks, scaling laws often follow power-law behavior [4], where performance metrics such as loss or error decrease predictably with increased model size or data. This relationship allows researchers to make informed decisions about resource allocation when training large-scale models. The power-law function can fit the validation performance curve and extrapolate it to larger data set sizes.

Model Size: As model size (Number of Parameters) increases, performance improves, but the improvement follows diminishing returns unless the dataset and compute scale appropriately.

Dataset Size: More complex tasks require larger datasets (Number of Samples). As models scale in size, they need more data to avoid overfitting and to continue improving performance. Tasks like machine translation or autonomous driving require vast amounts of data due to the complexity and variability of inputs.

Computational Power: Larger models and datasets naturally require more computational power. High complexity tasks, especially in reinforcement learning, can demand enormous computational resources to simulate environments and optimize policies.

Performance scales predictably with model size, dataset size, and compute. These relationships follow power laws over several orders of magnitude. As long as model size and dataset size are increased together, performance continues to improve.

Figure 2 shows a sketch power-law plot that breaks down

Aspect	Machine Learning (ML)	Deep Learning (DL)
Data Volume	Requires moderate datasets (hundreds to thousands of samples). Suitable for structured data.	Requires large datasets (thousands to millions of samples) to capture patterns, especially in unstructured data.
Feature Engineering	Relies on manual feature selection and domain expertise to identify predictive features.	Automatically extracts features directly from raw data, reducing the need for manual input.
Data Diversity	Moderate data diversity is needed; focus on specific features or categories.	High data diversity is essential to generalize effectively across different scenarios.
Preprocessing Effort	Requires extensive preprocessing (cleaning, structuring, handling missing values).	Requires less preprocessing; handles raw data but often needs normalization and augmentation.
Labeling Needs	Smaller datasets with labeled data suffice for most tasks.	Large volumes of labeled data are critical for supervised tasks like image or text recognition.
Sensitivity to Quality	Highly sensitive to quality; smaller datasets amplify the effect of errors or noise.	Moderately sensitive to quality but relies on large datasets to mitigate noisy data.

Table 2: Comparison of Machine Learning and Deep Learning regarding data requirements.

learning curve phases. The curve begins in the small data region, where models will struggle to learn from a small number of training samples. Here, models can only perform as well as "best" or "random" guessing. The middle portion of the learning curves is the power-law region, where each new training sample provides information that helps models improve predictions on previously unseen samples.

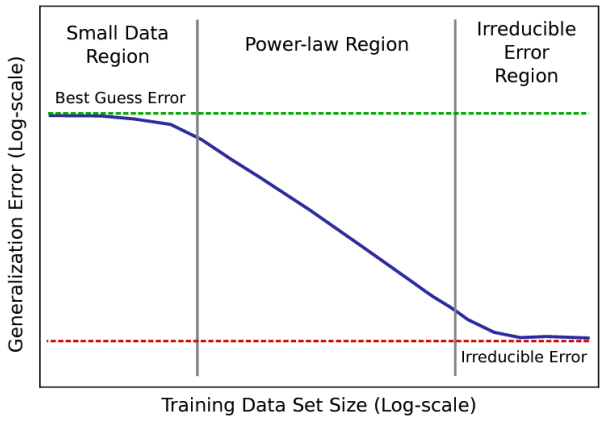


Figure 2: Sketch of power-law learning curves.

The power laws may apply or fail depending on the data distribution and underlying model complexity. Understanding these laws is crucial for optimizing the balance between computational expense and model performance [4].

Estimation of Data Requirements based on Scaling Laws

The amount of data required can vary significantly based on the complexity of the task and the AI methods applied. Overestimating or underestimating data requirements incurs substantial costs that could be avoided with an adequate budget.

Determining how much data is necessary to reach a target validation or test performance is a critical initial requirement when designing an AI solution. The amount of data needed can vary significantly depending on the complexity of the problem.

To address this, we categorize tasks into three levels—Low Complexity Tasks, Medium Complexity Tasks, and High Complexity Tasks—based on scaling laws and the relationship between task difficulty, data types, and model sizes, as outlined in Table 3.

Low Complexity Tasks: These tasks, like binary classification or simple regression, don't require massive models or datasets. A model with 10M to 100M parameters is often sufficient, and you can train them on smaller datasets using fewer computational resources.

Moderate Complexity Tasks: For tasks like multiclass classification or continuous control RL, you need models in the 100M to 500M parameter range, larger datasets, and computational resources in the range of hundreds of thousands of GPU hours.

High Complexity Tasks: Tasks such as machine translation, image segmentation, and autonomous driving need the largest models (1B to 10B parameters), huge datasets, and millions of GPU hours for training.

This requirement is determined by factors such as the size of the training data set, the learning algorithm applied, and the specific objectives of the application. By categorizing tasks into varying levels of complexity, this approach offers a structured framework for estimating data requirements and ensuring that model performance is optimized according to the demands of the task.

Task Complexity	Data Type	Small Model	Medium Model	Large Model
Low Complexity	Text	1,000 – 10,000	10,000 – 50,000	50,000 – 100,000
	Tabular	1,000 – 5,000	5,000 – 20,000	20,000 – 50,000
	Audio	1,000 – 5,000	5,000 – 20,000	20,000 – 50,000
	Environmental	10,000 – 50,000	50,000 – 100,000	100,000 – 500,000
Moderate Complexity	Text	10,000 – 50,000	50,000 – 200,000	200,000 – 1 million
	Tabular	5,000 – 20,000	20,000 – 100,000	100,000 – 500,000
	Audio	10,000 – 50,000	50,000 – 200,000	200,000 – 500,000
	Environmental	50,000 – 100,000	100,000 – 500,000	500,000 – 1 million
High Complexity	Text	100,000 – 500,000	500,000 – 1 million	1 million – 10 million
	Image	100,000 – 500,000	500,000 – 1 million	1 million – 10 million
	Time Series	50,000 – 200,000	200,000 – 1 million	1 million – 10 million
	Audio	100,000 – 500,000	500,000 – 1 million	1 million – 10 million
	Environmental	500,000 – 1 million	1 million – 10 million	10 million – 100 million

Table 3: Data Size Estimation Based on Task Complexity and Model Size.

2.3. Data Collection and Storage

Data collection and storage are fundamental to building effective machine learning systems. This involves understanding what type of data is required, determining where it can be sourced, and establishing how it will be collected. Proper storage solutions must also be planned to ensure scalability, security, and accessibility.

key questions that should be asked

- What data sources will be used, and how reliable and up-to-date are these sources?
- How well do I understand the characteristics of my data and the conditions under which they were collected?
- What potential downstream problems could result from using these specific data?
- How frequently does that data need to be updated to maintain the model's performance over time?
- Will the data be stored on-premises, in the cloud, or using a hybrid solution?
- Does the storage infrastructure support scalability as data volume grows?

Data Collection

The data collection phase is crucial for determining how much data is needed, how to reduce data dependency, and when data needs to be collected. The frequency of data collection ensures that that data is up-to-date, consistent, and accurate. Collecting data too infrequently risks missing significant changes, trends, or events that could impact analysis. Conversely, collecting data too frequently can lead to unnecessary costs, increased complexity, and heightened risks.

Manufacturing Data Sources

- **IoT Sensors**: Real-time data like temperature, pressure, and energy.
- **Production Logs**: Records of downtimes, production rates, and activities.
- **Quality Control**: Inspection data, including images and measurements.
- **ERP Systems**: Inventory, supply chain, and production planning data.
- **Operator Input**: Manual records or adjustments providing context to automated data.
- **Simulation Data**: Digital twins or synthetic data for scalable AI training.

To determine the optimal data collection frequency, consider the rate of change and level of detail required from the data source, as well as the expectations and needs of the destination. It's essential to weigh the trade-offs and costs associated with both high and low-frequency data collection. For applications where conditions evolve slowly, data collection may only need to happen quarterly or annually, followed by model updates. In fast-changing environments (e.g., manufacturing systems with frequent process changes), more frequent data collection—daily or weekly—may be necessary. Leveraging real-time data streams for ongoing updates can reduce the burden of large batch collections and ensure models remain up-to-date.

Data Storage

A robust data infrastructure must securely store all collected data, with security and accessibility as top priorities. Key steps include selecting the right storage system, defining data retention policies, and establishing backup and recovery processes to maintain data integrity and availability.

Depending on specific needs, storage options might include cloud or physical (on-site) servers. While on-site storage, or local storage, is often viewed as more secure, this is not always the case. Secure cloud storage with carefully defined access controls and regular backup protocols can safeguard data without the added responsibility of managing hardware and infrastructure.

Scaling data infrastructure is critical for handling the growing volume, variety, and velocity of data generated in modern manufacturing environments. This includes cloud storage, databases, and distributed systems that ensure reliability, scalability, and efficiency—essential requirements for any data storage solution (section 4.1).

Example

A manufacturing company utilizes a centralized platform to store and analyze data from multiple plants globally. The system needs to reliably handle critical data from thousands of sensors, scale with growing data volumes, and support efficient processing to generate actionable insights, such as optimizing energy consumption or enhancing production efficiency.

2.4. Data Quality and Analysis

Data analysis relies heavily on the quality of the data being used. High-quality data—accurate, complete, and consistent—is essential for generating meaningful insights and reliable results. Poor-quality data can lead to errors, biased models, and ineffective decision-making.

key questions that should be asked

- How does the model handle unseen or out-of-distribution data?
- How often should the data distribution be analyzed to detect drift?
- How often should the model be retrained, and on what criteria?
- What quality concerns or biases are present in my data?
- Is the model achieving its intended performance metrics on real-world data?

Data Quality

It is a critical determinant of the success of AI applications in manufacturing. High-quality data ensures that the patterns and insights derived from AI methodologies are accurate and reliable. Poor data quality can lead to inaccurate models, misleading results, and ultimately, faulty decision-making. Therefore, it is crucial to implement robust data quality assurance mechanisms to ensure the reliability and accuracy of AI applications.

Various problems can arise during data collection, leading to uncertainties and delays. These problems are shown in Table 4.

Factor	Description
Data Quality	Incomplete, erroneous, or duplicate data can distort analysis and require extensive cleaning and validation efforts.
Data Inconsistency	Data from different sources may be inconsistent or inaccurate, necessitating additional time to resolve discrepancies.
Data Noise	Irrelevant or erroneous data points that obscure meaningful patterns and reduce the clarity and reliability of the dataset.
Historical Data Accessibility	Delays can occur due to the lack of accessibility or poor documentation of historical data.
Real-time Data Collection	Implementing sensors and data pipelines for real-time data collection can be time-consuming and complex.
Data Granularity	Determining the appropriate level of data granularity is often uncertain and project-specific, impacting the time required for collection and processing.
Data Compatibility	Incompatible data formats and definitions can lead to time-consuming conversions and adjustments.
Technical Infrastructure	Inadequate or outdated infrastructure can slow down the data collection process.
Scalability and Flexibility	Systems that are not scalable or flexible enough to respond to changing requirements can introduce uncertainties in the timeline.
Time Management and Planning	Over-optimistic timelines and undetected issues can lead to delays and inaccurate results.

Table 4: Factors affecting the overall project timeline.

Addressing these uncertainties and implementing strategies to minimize the time required for data collection is

essential for efficient project management.

Maintaining high-quality data is a continuous challenge due to various issues that can arise at different stages of the AI lifecycle such as data drift, concept drift, bias, imbalance, and noise that can significantly affect the accuracy and generalizability of AI models.

Imbalanced Data Imbalanced Data occurs during model training, when the distribution of classes or categories in a dataset is significantly skewed, meaning one class is much more frequent than others. This imbalance can lead to biased models that perform well on the majority class but fail to accurately predict or recognize the minority class.

Data Biased Bias arises when the training data are not representative of the real-world distribution. This can lead to unfair or inaccurate predictions, especially if certain groups, conditions, or scenarios are underrepresented in the dataset. Furthermore, bias may not only exist in the training phase but can also emerge during deployment, as unseen biases in live data come into play, further complicating the performance and reliability of the AI system.

Data Drift It occurs when the statistical properties of the input data change over time after deployment. This means that the distributions, patterns, or underlying characteristics of the data shift, even though the relationship between inputs and outputs remains the same. Data drift might happen due to seasonal trends, changes in user behaviour, or external factors that impact data collection.

Concept Drift It happens during Post-deployment when the relationship between input data and the target variable changes. This is common in applications with evolving environments, where patterns that once indicated a certain behaviour no longer apply. Concept drift is more challenging to detect, as it requires monitoring performance metrics, rather than just input distributions. If the model's accuracy or other performance indicators begin to decline, this can signal concept drift, suggesting a need for model retraining or adjustment to reflect the new relationship.

Addressing these challenges proactively during data preparation and through continuous monitoring post-deployment is essential to maintaining high data quality and ensuring reliable AI performance.

Data Analysis

It is the systematic examination, transformation, and interpretation of data to extract meaningful information, identify patterns, and support informed decision-making. It involves employing various techniques and tools to discover insights, uncover trends, and answer questions about historical, current, and future events. By leveraging data analysis, organizations gain a deeper understanding of their operations, customers, and market environment, ultimately guiding strategic planning and optimizing performance.

When it comes to data analysis, there are generally four key types. Each differing in complexity, performance impact, and data requirements. Figure 3 illustrates the varying degrees of these four types, comparing the level of complexity involved with the value they add to the organization.

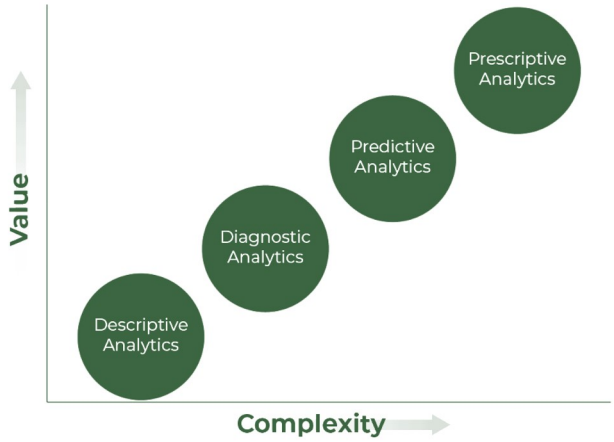


Figure 3: Data Analysis key types.

Descriptive Analysis: Summarizes and interprets historical data to highlight patterns, trends, and outcomes, effectively illustrating what has happened.

Diagnostic Analysis: Delves deeper into the factors behind observed results, aiming to determine why certain events took place or why certain patterns emerged.

Predictive Analysis: Uses historical data, statistical models, and machine learning algorithms to forecast future scenarios, enabling organizations to anticipate changes and prepare accordingly.

Prescriptive Analysis: Builds on the insights gained from other analysis types to suggest actionable recommendations, outlining the best course of action and helping decision-makers choose strategies with the highest probability of success.

The general principle is to start with the simplest category, Descriptive Analytics, which requires minimal data and provides basic insights into past events. As organizations progress to Diagnostic Analytics, Predictive Analytics, and finally Prescriptive Analytics, the complexity increases, requiring more advanced models, higher data quality, and larger volumes of data. This progression ensures a balance between achievable performance and growing data requirements while unlocking greater organizational value.

3. Solving the Data Bottleneck

An integrated approach to addressing the data bottleneck and associated risks is explored across the AI lifecycle. This includes strategies for overcoming data scarcity, ensuring high-quality datasets through preprocessing and cleaning, and managing imbalanced data to improve fairness and accuracy. Post-deployment monitoring is highlighted to address data changes and sustain model performance, enabling organizations to build reliable and adaptable AI systems.

3.1. Dealing with Data Scarcity

Data scarcity poses a significant challenge in AI projects, particularly when building models that require large, high-quality datasets. The need for additional data can arise at various stages of an AI project.

Problem Definition Phase: During the initial stages, when the scope of the problem is being defined, and it's unclear if the available data covers all relevant scenarios or variables. Additional data might be required to validate the problem scope and ensure representativeness for the AI model.

Data Collection and Preprocessing Phase: While gathering and cleaning data, gaps, biases, or imbalances in the dataset may be identified. More data may be needed to fill these gaps, address underrepresented classes, or ensure diversity and quality.

Model Training Phase: If the model underperforms due to insufficient training data, particularly for complex tasks or rare events. Additional data improves model generalization, accuracy, and robustness, especially for supervised learning tasks.

Validation and Testing Phase: During validation, when the test results indicate overfitting or an inability to generalize to unseen data. More diverse or representative data ensures better testing and reduces bias in evaluation.

Post-deployment Phase: Data drift or concept drift occurs due to changes in real-world conditions. Collecting updated real-time data ensures the model remains accurate and adapts to evolving scenarios.

To mitigate data scarcity, organizations can use strategies like data augmentation, transfer learning, design simulation experiments, domain nknowledge integration, and active learning to maximize the utility of available data [5]. These approaches help ensure AI models achieve robust performance, even when data is limited.

Data Augmentation

Data augmentation is a technique used to artificially increase the size and diversity of the training dataset without collecting new data. This is particularly valuable for small manufacturing companies, where data collection can be costly or time-consuming. By simulating real-world variability, augmentation helps models generalize better, improving performance in practical applications.

Basic data augmentation techniques involve simple geometric transformations [6]:

Rotation and Flipping: Rotating images by fixed angles or flipping them horizontally/vertically helps the model recognize products regardless of their orientation.

Scaling and Translation: Scaling changes the size of the product in the image, while translation shifts its position within the frame. These techniques introduce variability, enabling the model to learn invariant features across different scales and positions.

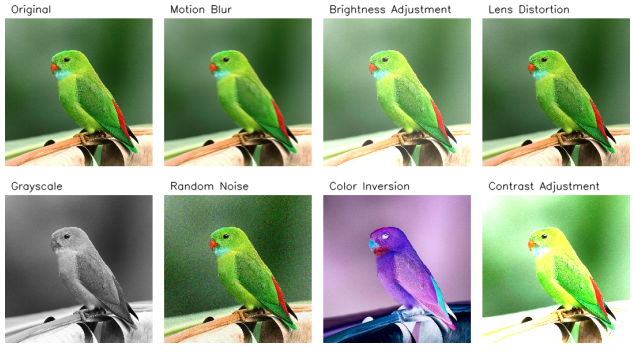


Figure 4: The effects of common data augmentation methods.

More sophisticated data augmentation methods can significantly enhance model performance, especially when dealing with limited data:

Generative Adversarial Networks (GANs): generate highly realistic synthetic data. In manufacturing, GANs can create images of rare defects, enriching the training dataset and

improving the model's ability to detect such defects.

Mixup and CutMix

- Mixup: Generates new training examples by combining two images and their corresponding labels through linear interpolation, encouraging the model to learn smoother decision boundaries.
- CutMix: Involves cutting out a region from one image and pasting it onto another, which increases data diversity and forces the model to focus on less obvious features like texture or shape.

Transfer learning

Transfer learning involves leveraging pre-trained models, typically trained on large, general-purpose datasets, and fine-tuning them for a specific task using a smaller dataset. For small manufacturing companies, this approach is invaluable as it allows them to benefit from the extensive knowledge embedded in these pre-trained models without requiring large amounts of data.

Example

A small textile manufacturer used a pre-trained model designed for general image recognition and fine-tuned it to detect fabric defects. By leveraging the model's existing knowledge of textures and patterns, they significantly reduced the time and data needed to achieve high accuracy in defect detection.

One popular model for transfer learning is VGG16, a deep convolutional network pre-trained on the ImageNet dataset. VGG16 is particularly suitable for image-based tasks, such as defect detection, due to its ability to capture intricate visual features.

Layer-wise Freezing and Unfreezing: A typical approach starts by freezing the lower layers of the pre-trained model, which capture general features like edges and textures. The final layers, which are more specialized, are fine-tuned on the new dataset. Gradually unfreezing earlier layers allows the model to adjust to the specific characteristics of the new data, such as unique textures or shapes found in defect images.

Domain Adaptation Techniques: Domain adaptation ensures that the pre-trained model aligns with the specific characteristics of the manufacturing data. Fine-tuning the model to recognize particular types of defects or textures ensures it is not just a general-purpose classifier but optimized for the specific needs of the manufacturing environment.

Design Simulation Experiments

Simulation experiments serve as a cornerstone in diverse disciplines, providing a platform to explore, optimize, and predict behaviors of complex systems that might be challenging to investigate in the real world. From physics to social sciences, these experiments offer a way to navigate "what if" scenarios, validate theories, and guide decision-making processes.

Designing a simulation experiment allows manufacturers to evaluate the behaviour of machine learning models in a controlled, virtual environment. By doing so, manufacturers can:

- Test how models will respond to real-world scenarios without disrupting actual production.
- Identify and mitigate potential risks or failure points in the ML models before full-scale deployment.
- Fine-tune ML models for specific operational conditions and edge cases that might not be present in the training data.
- Ensure the models' predictions or optimizations are consistent, reliable, and beneficial under various production conditions.

Real-Time Data Collection

Real-time data collection has become crucial in rapidly changing environments.

Streaming data: Leveraging APIs to continuously gather data from various sources.

Real-Time Logging: Continuously recording system logs and operational data to monitor performance, identify bottlenecks, and detect anomalies as they occur.

Sensor Networks: Using interconnected sensors to capture physical data in real-time, such as temperature, vibration, energy usage, and production efficiency, particularly in manufacturing and industrial settings.

User interactions: Capturing real-time data based on user behavior within applications, offering valuable insights into user preferences and emerging trends.

This method enables datasets to be continuously updated, ensuring AI models are trained with the most up-to-date information available.

Integrating Domain Knowledge

One effective approach for reducing sample complexity is integrating prior expert knowledge. In classical engineering, established theories and expert knowledge provide a solid foundation for system understanding. Integrating this domain knowledge into AI methods can significantly reduce the amount of data required, boost system reliability, and even enable extrapolation beyond known data points [7].

Integrating domain knowledge often requires additional effort to blend standard AI methods with external knowledge sources and engineering models, sometimes in mathematical forms like differential equations.

3.2. Data Preprocessing and Cleaning

In today's data-driven landscape, the success of any AI system relies significantly on the quality of its training data. AI data preparation—the process of cleaning, organizing, and structuring raw data—forms the crucial groundwork that allows AI models to deliver accurate and dependable insights. Without well-prepared data, even the most advanced algorithms can fall short, leading to misleading results and overlooked opportunities.

The overall process can be summarized as shown in Figure 5.

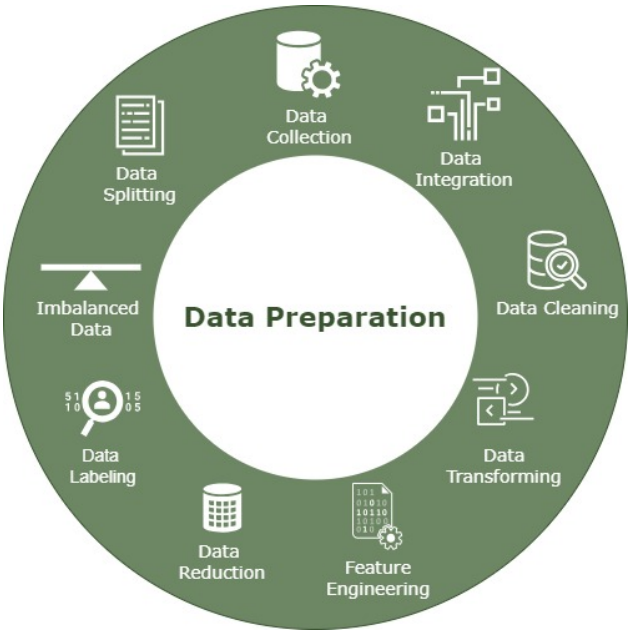


Figure 5: Data Preprocessing and Cleaning.

Data Integration

Integrating data from multiple sources is essential for comprehensive analysis but can be a complex task. It involves combining structured, semi-structured, and unstructured data from diverse systems while addressing challenges like format incompatibilities, data redundancy, and inconsistencies. Ensuring compatibility and consistency across datasets is critical for creating a unified view, enabling accurate insights, and improving the reliability of AI models.

Example

Combining simulated data with real-world data to enhance predictive maintenance models, ensuring the simulated data accurately reflects real conditions.

Data Cleaning

It involves identifying and correcting errors, inconsistencies, and inaccuracies to ensure data quality and reliability.

Handling Missing Values: Depending on the nature of the data and the problem, you can impute missing values using techniques like mean, median, mode, or more advanced methods such as k-NN imputation.

Removing Duplicates: Duplicate entries can introduce bias, so they should be identified and removed.

Outlier Detection and Treatment: Outliers can skew model results. Depending on the context, outliers can be removed, capped, or adjusted.

Data Transforming

This could include normalization, aggregation, or other techniques to make the data suitable for modelling.

Normalization: Scaling features to have a mean of 0 and a standard deviation of 1. Useful for algorithms that are sensitive to feature scales.

Standardization: Scaling features to lie between a given minimum and maximum value, often between 0 and 1.

One-hot Encoding: Converting categorical variables into a form that could be provided to machine learning algorithms to do a better job in prediction.

Feature Engineering

It is the process of transforming and creating features from raw data to improve the performance of machine learning models. It bridges the gap between raw data and model training by enhancing the dataset's relevance and predictive power.

- Generating new features by combining or deriving insights from existing ones.
- Creating interaction terms that capture the combined effect of two or more variables.
- Applying domain-specific knowledge.

Data Reduction

It involves techniques to reduce the volume of data while preserving its meaningful characteristics. It is essential for improving computational efficiency, reducing storage requirements, and enhancing model performance by focusing on the most relevant information.

Dimensionality Reduction: Techniques like Principal Component Analysis (PCA) or autoencoders can be employed to reduce the number of features if the dataset is too large, ensuring faster training without losing much information.

Feature Selection: This involves selecting the most critical features that contribute to the prediction. Techniques can be as simple as correlation matrices or more advanced ones like recursive feature elimination.

Data Sampling: Selecting a representative subset of the data, such as random sampling or stratified sampling, to reduce size without losing diversity.

Data Labeling and Annotation

It is a critical step in supervised and semi-supervised machine learning, where human annotators or automated systems assign meaningful labels to raw data to make it usable for model training. Effective data labeling strategies ensure high-quality and scalable annotations.

Automated Labeling: Using algorithms to label data automatically, often employed in situations where patterns are straightforward such as Rule-based systems, pre-trained models, or simple heuristics.

Crowdsourcing: Leveraging platforms like Amazon Mechanical Turk to distribute labeling tasks to a large pool of non-expert human annotators.

Active Learning: A strategy where a model identifies the most informative data points to be labeled, often focusing on data points it is most uncertain about.

Weak Supervision: Using noisy, limited, or imprecise sources such as heuristics, data programming, or knowledge bases to generate labels.

3.3. Dealing with Imbalanced Data

Unbalanced data occurs when certain classes are significantly underrepresented compared to others. In manufacturing, this imbalance often results in AI models that perform poorly on the minority class, which may represent critical defects. This can lead to undetected quality issues, increased waste, and potential financial losses.

Example

A small electronics manufacturer produces 10,000 circuit boards daily, with only 50 identified as defective. Initial AI models were trained on this highly unbalanced dataset, leading to poor performance in detecting defects. Consequently, some defective boards passed through quality checks unnoticed, causing significant customer dissatisfaction and returns.

Techniques for Addressing Imbalanced Data

Several techniques can be employed to mitigate the effects of unbalanced data.

Resampling Techniques

- Over-sampling: This technique involves generating additional synthetic samples for the minority class to balance the dataset. One popular method is the Synthetic Minority Over-sampling Technique (SMOTE), which generates new instances by interpolating between existing ones. While this reduces overfitting, it can introduce noise if not carefully managed.
- Under-sampling: This method reduces the number of samples in the majority class by selectively removing instances. While it balances the dataset, there is a risk of losing valuable information, potentially reducing the overall model performance.

Ensemble Learning Methods

- Balanced Random Forest: A variant of the random forest algorithm, this method applies balanced bootstrap sampling to ensure that each decision tree is trained on a more balanced subset of data, improving performance on the minority class.
- EasyEnsemble: This technique creates multiple balanced subsets from the majority class and trains weak learners on each subset combined with the minority class. The final model aggregates these learners, yielding robust performance on both majority and minority classes.

Cost-sensitive learning It addresses class imbalance by assigning higher penalties to the misclassification of minority classes. In manufacturing, this approach ensures that the model focuses more on detecting defects, even if they are rare, by incorporating the costs associated with different types of errors.

Balancing Complexity

When training an AI system, it is essential to strike a balance between two undesirable extremes: underfitting and overfitting. Both can significantly impact the performance and generalization ability of the ML model [8].

Underfitting It occurs when the model is not trained for a sufficient number of epochs or lacks the complexity needed to capture the underlying patterns in the data. Consequently, the model fails to adequately learn from the training data, leading to suboptimal predictions and poor performance on both the training and validation datasets. This scenario often arises from insufficient training time, overly simplistic model architectures, or inadequate feature selection.

Overfitting It happens when the model is trained excessively or is too complex relative to the data. In this case, the model fits the training data too closely, capturing not only the underlying patterns but also noise and random fluctuations. As a result, while the model may perform exceptionally well on the training dataset, it struggles to generalize to new, unseen data, yielding poor predictions. Overfitting highlights the importance of focusing on the generalization capacity of ML models, as their primary goal is to learn broad concepts rather than memorizing specific examples.

Figure 6 is an illustration of overfitting and underfitting. As the number of weight update iterations grows, the model progressively learns the training data, leading to improved generalization performance. However, a critical "breaking point" or inflection point is reached, beyond which additional training increases the model's accuracy on the training data but adversely affects its generalization ability.

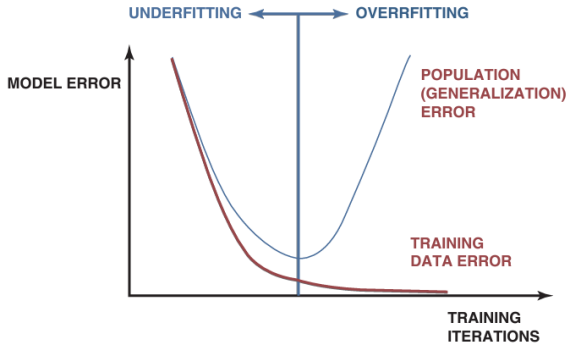


Figure 6: Illustration of overfitting and underfitting.

To mitigate these issues:

- Optimize the model's performance by adjusting hyperparameters using techniques like grid search or random search. These methods explore different parameter combinations and select the ones that yield the best performance based on the validation data.
- Early stopping approach: monitoring the model's performance on a validation dataset during training and halting the training process when the validation error stops improving, thereby avoiding overfitting.
- Using regularization methods such as L1/L2 penalties by adding constraints to the optimization process.

$$\text{Loss}_{\text{L1}} = \lambda \sum_{i=1}^{n} |\theta_i|$$

$$\text{Loss}_{\text{L2}} = \lambda \sum_{i=1}^{n} \theta_i^2$$

where λ is the regularization parameter controlling the penalty strength, θ_i are the model coefficients.

$$\text{Loss}_{\text{ElasticNet}} = \lambda_1 \sum_{i=1}^{n} |\theta_i| + \lambda_2 \sum_{i=1}^{n} \theta_i^2$$

where λ_1 controls the L1 penalty, λ_2 controls the L2 penalty. These penalty terms are added to the main loss function (e.g., Mean Squared Error, Cross-Entropy Loss) to improve generalization and ensure better model performance.

3.4. Model Verification and Validation

Once a model has been chosen, verifying and validating its reliability and effectiveness becomes essential. Verification and Validation serve as checkpoints that ensure a model accurately represents the phenomena it aims to simulate and performs as expected in real-world conditions.

Data splitting is a fundamental step in this process, where the dataset is divided into training, validation, and test sets. The training set is used to build the model, the validation set is used for verification, and the test set is reserved for final validation to assess the model's generalizability. High-quality, representative data is crucial during data splitting to ensure that verification and validation processes yield meaningful results. Resolving data issues during verification ensures the model functions correctly before validation, where its true performance is tested on unseen data. This structured approach ensures that the model is robust, reliable, and ready for real-world application.

Data Splitting

Data is typically split into Training, Validation, and Test subsets. The training set is for training the model, the validation set is for tuning hyperparameters, and the test set is for evaluating model performance.

Holdout Method: The dataset is randomly divided into three subsets: training, validation, and testing. This simple approach provides an efficient way to evaluate model performance but may lead to variability depending on the random split.

K-Fold Cross-Validation: The dataset is split into k subsets (folds). The model is trained k times, each time using $k - 1$ folds for training and the remaining fold for testing. This method reduces variability by averaging performance across all folds, providing a more robust evaluation.

Bootstrapping: A resampling technique where multiple samples are drawn with replacement from the dataset to train and evaluate the model. Bootstrapping helps estimate model performance when the dataset is small or limited, capturing variability by creating diverse training sets.

Model Verification

To ensure the model is error-free and functions as intended, aligned with its theoretical or conceptual foundations.

Data Cleaning and Preprocessing: The verification phase heavily relies on clean, well-preprocessed data to confirm the model's functionality.

Data Consistency Checks: Consistency checks ensure that all data conforms to expected formats, ranges, and structures. Anomalies, like sudden spikes in time-series data, might indicate data recording issues rather than true events.

Data Transformation Validation: For models that require feature transformations, it's essential to verify these transformations produce consistent outputs, especially if transformations are applied dynamically.

Model Validation

To determine if the model's outputs are valid and useful for its intended purpose. Validation checks the model's performance on real-world data, making data structure, quantity, and diversity foundational to assessing the model's robustness.

Validation Data Representativeness: Validation datasets must cover the full range of scenarios the model will face post-deployment. Underrepresented groups in the validation data can lead to biased predictions or erroneous outputs.

Cross-Validation with Data Diversity: Cross-validation relies on splitting data into training and validation subsets. Ensuring that each subset is representative prevents overfitting and promotes generalization across all data segments.

Data Quantity for Reliability: Sufficient data is needed for the model to "see" enough variations in scenarios. Limited or narrow data can lead to underfitting, where the model fails to learn relevant patterns, or overfitting, where the model performs well on training data but poorly on new data.

Performance Evaluation

It is essential to monitor the ML model's performance against key metrics.

Prediction Accuracy: How accurately does the model predict machine failures or defects?

> In classification tasks, metrics like accuracy, precision, recall, F1-score, or AUC-ROC are commonly used. For regression problems, mean squared error (MSE) or mean absolute error (MAE) may be more appropriate.

Optimization Effectiveness: Is the ML model effectively optimizing production parameters, reducing waste, or lowering energy consumption?

Response Time: How quickly does the model adjust its predictions or recommendations based on new data?

Robustness: How does the model perform under different operating conditions or edge cases? Does it generalize well to unseen data or scenarios?

Return on Investment (ROI): Does the simulation show that the ML model delivers tangible benefits, such as reduced downtime, improved product quality, or enhanced efficiency?

Sensitivity Analysis: This tests the model's stability against variations in input data, such as noise or outliers. It's vital to ensure that the model performs consistently under various conditions.

3.5. Continual Learning

As AI models are deployed in dynamic environments, data patterns and requirements can evolve over time. To maintain model relevance and accuracy, it is crucial to establish continuous monitoring and adaptive processes, we proposed a continual learning (post deployment) cycle shown in Figure 7.

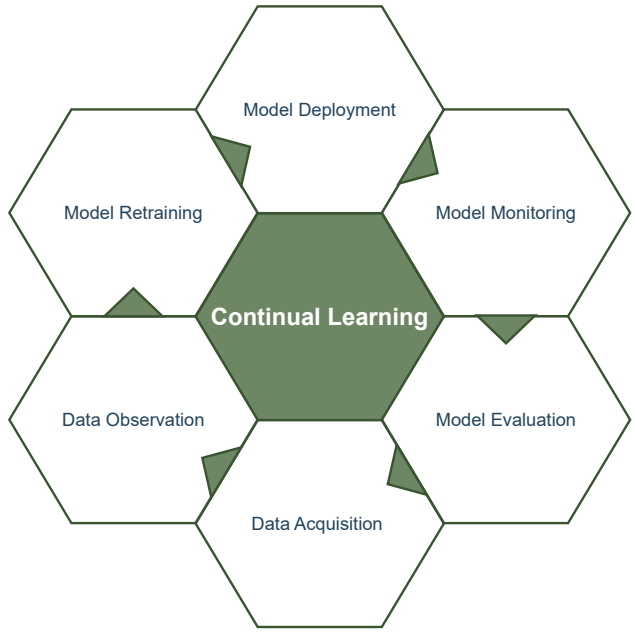


Figure 7: Continual Learning Cycle.

Model Deployment and Monitoring

An effective deployment includes the following components, listed in order of increasing implementation complexity.

Continuous Integration/Continuous Deployment (CI/CD): Setting up a strong CI/CD pipeline is essential to enable rapid rollback to a previous model version if needed.

Model Monitoring: Monitoring the model's performance and behaviour is as valuable as monitoring messages and API requests. Together, these insights allow your team to understand the model's behavior and quickly diagnose issues.

A/B Testing: A hands-on method for assessing model performance across different conditions is A/B testing.

CI/CD with Humans-in-the-Loop: This approach includes testing the model on live samples before deploying it in

production, which helps to reduce errors in real-world settings. It's a valuable method for verifying models against known issues, as seen in self-driving AI, where each known challenge undergoes extensive human validation.

Data Acquisition and Observation

After deployment, acquiring new data is essential to ensure AI models remain accurate and relevant in dynamic environments. New data can capture changes in patterns, user behavior, or system conditions that were not present during initial training.

Once a robust model monitoring system is in place, detecting data drift and performance degradation becomes more effective.

A common approach is to use monitoring dashboards that visualize key metrics over time.

Stability Metrics: Data Drift, Concept Drift, Model Drift.

Performance Metrics: Accuracy, Precision and Recall, AUC-ROC, F1 Score, MAE and MSE.

Operations Metrics: Memory, compute resources, Latency, Throughput, server load.

Input metrics: Model input distribution.

Continuous data acquisition allows for monitoring Data Drift, bias monitoring.

Model Monitoring for Data Changes Implement systems to track your models' performance, keeping an eye on key metrics and identifying signs of performance degradation or concept drift, where the underlying data distribution shifts. Setting up thresholds and alerts helps in the early detection of these issues.

Continuous Bias Monitoring It is essential to ensure AI models remain fair and unbiased post-deployment. Bias can emerge over time due to changes in data distributions, shifts in user behavior, or the introduction of new edge cases.

- Diverse and Representative Training Data:
  - Ensure that the data used for training AI models reflects the full diversity of the population or scenarios the AI system will encounter.
  - Conduct regular audits of the datasets to identify gaps or imbalances and take corrective action, such as supplementing data from diverse demographics or environments.
- Bias Detection Tools and Techniques:
  - Use statistical tools and algorithms designed to detect bias in data and model outcomes. These tools can help identify instances where the model's predictions disproportionately favor or disadvantage specific groups.
  - Implement techniques like fairness metrics and adversarial testing to ensure the model treats different demographic groups equitably.

Data-Driven Retraining and Model Updates

Regularly retrain your models using fresh data or update them to reflect changes in the data distribution or evolving problem context. This ensures that the models continue to make accurate predictions based on the most current data.

Iterative Improvement with New Data and Feedback Continuously enhance your models by incorporating feedback from real-world usage, applying domain-specific insights, and exploring new algorithmic techniques. This iterative process ensures that your models adapt and improve alongside changing data and business needs.

Monitoring data post-deployment is essential to ensure the AI model continues to perform reliably in real-world conditions.

Model Interpretability and Transparency

Model Interpretability and Transparency are critical aspects of deploying AI systems, especially in applications where understanding how a model makes decisions is essential.

- Build models that are interpretable, allowing stakeholders to understand how predictions are made and why. This is especially important in high-stakes applications, such as healthcare or finance, where decisions have significant real-world impacts.
- Provide clear documentation outlining the model's design, data sources, and potential biases, so that users can make informed decisions.

Effectively solving the data bottleneck is critical to the success of AI systems. Addressing issues like data scarcity, imbalances, and quality concerns not only improves model performance but also builds a foundation for long-term adaptability. Preprocessing and cleaning ensure data consistency, while strategies for post-deployment monitoring safeguard models against shifts in data patterns. By prioritizing these practices, organizations can confidently deploy AI systems that remain reliable, accurate, and aligned with evolving real-world demands.

4. Managing the Data Bottleneck

A robust data bottleneck management solution empowers organizations to intelligently unify their data, enabling better access, trust, and control. This capability is essential for achieving key business objectives, as every effort to enhance customer experience, streamline operations, or drive organizational transformation relies heavily on the effective use of data.

4.1. Data Management

Data management refers to designing and implementing the frameworks, standards, and guidelines necessary to address an organization's entire data lifecycle needs. Establishing these structures is essential for understanding and analyzing complex, large-scale data environments. By treating data as a critical business asset, organizations recognize the importance of effectively managing it. Data management involves the systematic collection, organization, control, and accessibility of data to enhance productivity, efficiency, and informed decision-making [9].

This process encompasses a wide range of tasks and procedures as shown in Figure 8.

Figure 8: The Process of Data Management.

Data Architecture Defines the blueprint for data flow, storage, and access, ensuring that systems, processes, and policies align to meet organizational needs.

Data Warehousing Centralize all data sources, offering a single, trusted environment from which analytical insights can be derived. By consolidating disparate data into one repository, they create a clear path to effective data analysis.

Data Context It refers to the circumstances, conditions, and metadata surrounding the data's collection, including its source, purpose, and any factors influencing its quality. Without context, critical patterns or anomalies may be misinterpreted, leading to flawed insights and decisions. Incorporating metadata, such as timestamps, environmental conditions, or operational settings, provides the necessary context to ensure that AI models are trained and deployed with a comprehensive understanding of the data, enhancing accuracy and reliability.

Data Versioning It tracks and manages changes to datasets over time, ensuring each iteration is stored, labeled, and retrievable. This allows rolling back to previous versions, comparing changes, and maintaining an auditable history. It is crucial for analytics, machine learning, and compliance, enabling validation, reproducibility, and accountability. By ensuring traceability and structure, data versioning enhances data quality, collaboration, and decision-making.

Data Storage Solutions Efficient data storage is critical in the AI workflow, ensuring that preprocessed data is securely stored for analysis and model training. Scalable and flexible storage solutions manage large volumes of diverse data generated by AI applications, supporting a range of tasks from simple analytics to complex machine learning models.

As shown in Table 5, different storage solutions suit various data management needs. Choosing the appropriate storage system impacts the efficiency, reliability, and cost-effectiveness of AI workflows. Aligning the storage solution with the AI application's specific requirements—such as data size, structure, access frequency, and compliance—maximizes data utility while minimizing bottlenecks.

Storage Solution	Description	Use Cases
SQL Databases	Provide structured storage with well-defined schemas and powerful querying capabilities.	Transactional data, relational databases.
NoSQL Databases	Flexible storage for unstructured or semi-structured data, including document and graph stores.	Horizontal scalability and handling diverse data types (Social networks, content management).
Data Lakes	Centralized repositories that store raw data in its native format, whether structured, semi-structured, or unstructured.	Big data analytics, ML workflows.
Cloud-Based Storage	Storage Scalable solutions provided by cloud platforms with global accessibility.	AI workflows with features like versioning, backup, and disaster recovery.
On-Premises Storage	Physical storage managed within an organization, ensuring full control over security.	Sensitive data storage, compliance.
Hybrid Storage	Combines on-premises and cloud storage for flexibility.	Store sensitive data locally while using the cloud for scalability and advanced analytics.

Table 5: Overview of data storage solutions adapted from [10].

Data Catalogs It manages metadata to create a complete picture of the data. They consolidate data dictionaries, define data elements, and centralize business rules, governance policies, and glossaries. This streamlines data discovery, enhances collaboration, and ensures access to reliable data assets. By improving metadata management, organizations foster a transparent and connected data environment, reducing complexity and empowering stakeholders to use data effectively.

4.2. Data Governance and Usage Rights

Data governance establishes the framework for managing data quality, security, and compliance, while usage rights define who can access, use, and share data. Together, they ensure data is used responsibly and ethically.

key questions that should be asked

- **Ownership**: Who is responsible for managing the data?
- **Access**: Who is authorized to access specific datasets?
- **Security**: What measures are in place to safeguard data and ensure privacy?
- **Compliance**: How much of the organization's data adheres to current regulations?
- **Approval**: Which data sources are verified and approved for use?

Governance policies include managing data ownership, defining roles, and ensuring regulatory compliance. Usage rights specify permissions and restrictions, addressing privacy concerns and legal requirements. Effective data governance and clear usage rights promote trust, mitigate risks, and maximize the value of data assets [11].

Data Security and Compliance They involve categorizing data sources by their risk levels and establishing secure access mechanisms. This ensures a balance between maintaining robust security protocols and enabling seamless user interactions with the data.

Data Stewardship It involves monitoring how teams utilize data sources, with stewards taking the lead in promoting best practices. Their role includes ensuring data access aligns with security and quality standards while fostering responsible data usage.

Data Transparency It is vital for effective data governance. All processes and procedures should operate within a framework that allows analysts and business users to clearly understand the origins of their data and be aware of any special considerations or limitations associated with the data. By implementing transparent governance models, organizations empower users to confidently leverage data for meaningful insights while maintaining compliance and security.

Through strategic investment in data management and governance, organizations can unlock the full potential of their AI initiatives. By addressing architecture, governance, storage, and operational practices, organizations can create a data environment that is both robust and adaptable.

5. Use Case: Wear Condition Detection of Ball Screw Drive Surfaces

In the domain of production, one significant area of focus in the application of AI is predictive maintenance. This topic encompasses the monitoring and evaluation of the wear condition of a given component, as well as the subsequent prediction of the point in time at which the component will become unable to fulfill its intended function. In the context of machine tools, the ball screw drive (BSD) plays a central role in achieving high-precision production of workpieces. However, mechanical failures, such as pittings on the surface of the BSD, frequently occur, resulting in unanticipated machine downtime [12]. It is therefore desirable to implement a predictive maintenance method capable of detecting these pitting defects and, moreover, planning a change of the BSD with a minimum of machine downtime. A significant challenge arises from the often limited data available in production environments, which are necessary for training AI applications. This problem is addressed in the following section, which outlines a solution based on the CRISP-DM methodology. Most of the following content is based on the dissertation of Tobias Schlagenhauf [13] and can be found in much more detail and depth in the cited document.

5.1. Business Understanding

A total of 38 % of machine downtime among machine tools is attributable to failure of machine axes, of which 38 % is also attributed to BSD-related causes [12]. Figure 9 illustrates the composition of a BSD. When the spindle is rotated by an electric engine, the nut will undergo a translational movement. The two components are connected via the balls located within the nut. A deficiency in lubrication results in dry contact between the balls and the surface of the spindle, leading to the formation of pittings on the surface. Consequently, the precision of the machine is compromised, necessitating an unplanned machine downtime to replace the ball screw drive. To address the issue of unplanned machine downtime, it is essential to develop a method for monitoring and quantifying the wear on the spindle surface. Therefore, the initial step is to implement a measurement system that enables the monitoring of sensor signals during the normal operation of the machine, which are correlated with the surface wear condition of the BSD. The data should then be used to develop a supervised artificial intelligence model for the purpose of detecting pittings on the spindle surface. Consequently, pretrained networks such as GoogLeNet and those developed in-house should be tested and evaluated. Due to the necessity of a substantial amount of labeled data, data augmentation (see section 3.1) should be employed to achieve enhanced detection accuracy with a reduced data set. The objective is to develop an AI algorithm that attains a detection accuracy of 99 % with a feasible amount of labeled data.

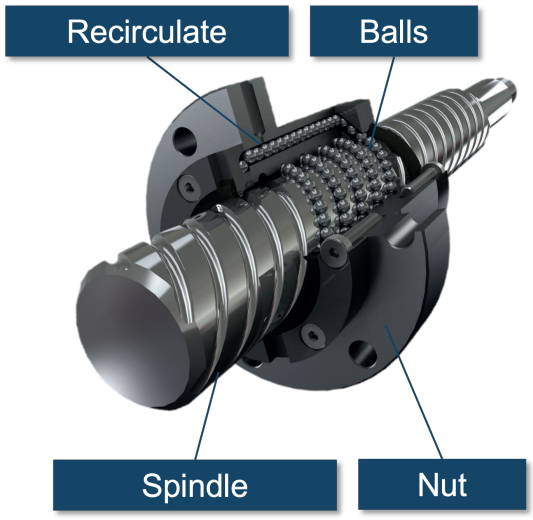


Figure 9: Composition of a ball screw drive.

5.2. Data Understanding

To detect pittings on the spindle surface, several correlated signals may be utilized, including acceleration measurements taken on the nut acoustic emission measurements taken on the nut, and current measurements taken from the electrical drive engine. However, it should be noted that these signals only provide indirect representations of the spindle surface wear condition. To obtain direct representations of the pittings, a camera-based measuring system is necessary, as illustrated in Figure 10. With such a system, it is possible to take images directly of the spindle surface including the pittings, as shown in Figure 11. To generate a large dataset with sufficient data examples of wear and depict the complete lifetime of BSD the sensor-system

was integrated multiple times on a test-bench specifically designed for lifetime tests of BSD. This test-bench allows to continuously move up to five BSD under significant load to actively induce wear. It has to be noted, that one such test run still takes 40 days of continuous BSD movement. During this time, the sensor-system automatically images the whole BSD every four hours [13]. The resulting dataset needs to be further prepared to be useful in the development of a wear-detection system. This will be discussed in the following section.

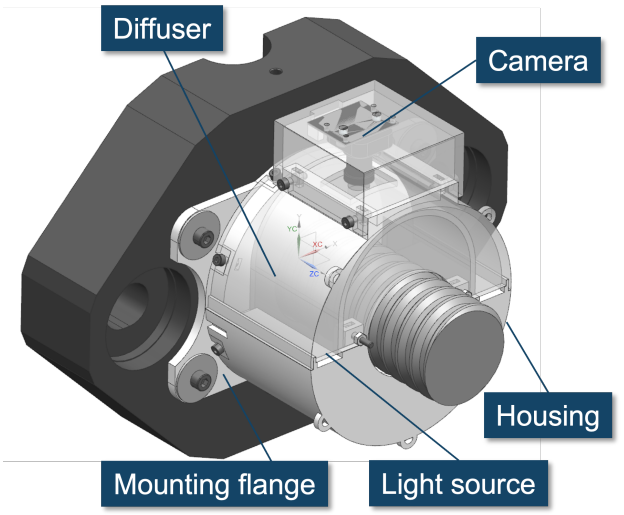


Figure 10: BSDcam – Camera system to make pictures of the spindle surface directly mounted on the nut.

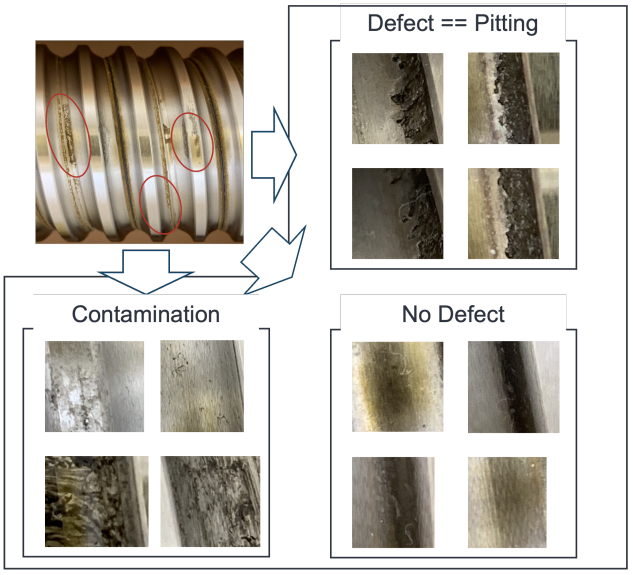


Figure 11: Example images from the dataset, showing contamination, pittings and no defect.

5.3. Data Preparation

To prepare the generated dataset for the training of a pitting classification system the smaller sections of 150x150 pixels were cropped from the original high resolution images provided by the camera system. The size was chosen to adequately represent both small and large scale damage. The cropped image segments were manually labeled by assigning them either to the class P (Positive for Pitting) or N (No Defect). This resulted in a Dataset of 21853 labeled images evenly distribute between P (10778 images) and N (11075 images). Additionally the dataset includes very diverse images of all possible BSD conditions, especially regarding contamination. Some examples can be seen in Figure 11. The complete dataset was published in [14] to contribute to the scientific community and ease the scarcity of data for similar use-cases.

To further supplement the dataset, data-augmentation techniques can be applied. By artificially increasing the size and diversity of the dataset, the models robustness and detection accuracy can be enhanced. In this use case, domain-specific transformations were applied to represent realistic environmental and operational variances. These transformations included slight rotations to simulate changes in thread pitch, perspective transformations for camera angle variances, and the addition of noise and blurring to mimic contamination and particle interference. Brightness, contrast, and color variations were adjusted to mimic changes in lighting conditions [13]. How this affected the results will be discussed in the following section.

5.4. Modeling

To automate image classification deep-learning based approaches, convolutional neural networks (CNN) are the current state of the art. Besides the quality of the dataset used to train these models their architecture is of critical importance for the classification accuracy. In the context of the use-case presented in this section multiple state of the art architectures were evaluated and a Design-of-Experiments (DoE) study was carried out to test 486 different CNN-architectures. Additionally it was investigated how the data-augmentation mentioned above affects the results. Schlagenhauf [13] was able to achieve good results with state of the art architectures (especially GoogLeNet) but could increase the reached accuracy again using a custom architecture from the parameter study. The performance of the best model was further increased by using the augmented data, even though the accuracy already reached values greater than 99 % as shown in Figure 12.

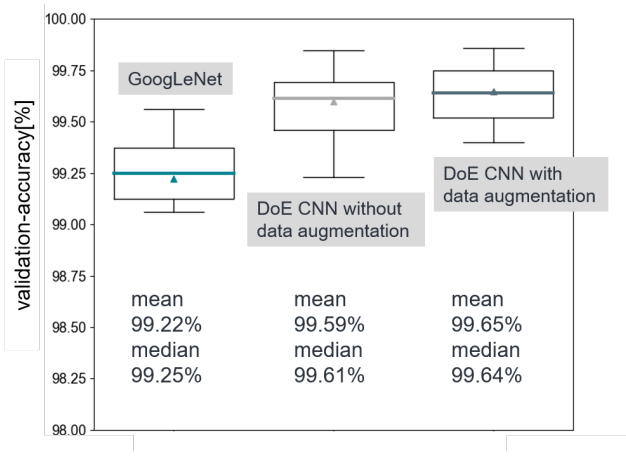


Figure 12: Accuracy results for the described use-case.

5.5. Validation

During the development of models as described in this use-case it is important to periodically validate the systems performance. Besides setting aside a portion of the training data for validation it is also possible to check which areas of an image were most important for the classification of an image as showing pittings. Doing so ensures, that the model actually learns to extract features that are specific to pittings, ensuring good performance on new image data. This can be done using a so called heat-map-approach ([13] based on [15]). This calculates gradients of the CNN and visualizes them in a heat-map overlay of the images, thereby highlighting areas that were particularly important for the classification. Figure 13 shows this for the example use-case.

Figure 13: Relevance of image areas for the classification of pittings.

5.6. Deployment

Deploying a system as described above involves integration both hard- and software-components into a machine tool. This means making sure, that the sensor-system is sufficiently shielded against fluids and chips present in the machine-tool environment. Further, it is necessary to connect the system to the machine tool controller. While the actual classification requires much less processing resources than the training, it still exceeds the capabilities of most control units, meaning additional edge-computing hardware is necessary. This still needs to be connected to the controller to trigger measurements and feed-back results to the controller and the user.

Since the use-case is a long-term application it is also necessary to periodically check the quality of current image data. There are a number of factors, that might affect image quality. The camera sensor might degrade, contaminants might get on the lens or lighting conditions on the shop floor might change with the time of day or even the time of year. Only continuously monitoring the data quality can ensure the peak performance of the described system.

Conclusion

With the camera system mounted on the nut of a BSD, it is possible to directly capture the wear conditions of a BSD surface. A data augmentation based extended image dataset recorded with this system can then be used to train different CNN architectures. Tests have shown that a GoogLeNet architecture results in a validation accuracy of greater than 99 %. However, using a custom architecture resulting from a parameter study, this accuracy could be increased even further. Validation also showed that the network calculates its predictions based on the pitting-relevant parts of the image. Based on such a system, fully automated and highly accurate wear condition monitoring could be implemented in the axes of a machine tool.

References

[1] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a standard process model for data mining". In: ***Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining***. Vol. 1. Manchester. 2000, pp. 29–39.

[2] Constanze Hasterok and Janina Stompe. "PAISE®–process model for AI systems engineering". In: ***at-Automatisierungstechnik*** 70.9 (2022), pp. 777–786.

[3] Andrew Ng. "MLOps: From model-centric to data-centric AI". In: ***DeepLearning. AI https://www. deeplearning. ai/wp-content/uploads/2021/06/MLOps-From-Model-centric-to-Data-centric-AI. pdf*** (2021).

[4] Joel Hestness et al. "Deep learning scaling is predictable, empirically". In: ***arXiv preprint arXiv:1712.00409*** (2017).

[5] Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade. "A review: Data pre-processing and data augmentation techniques". In: ***Global Transitions Proceedings*** 3.1 (2022), pp. 91–99.

[6] Alhassan Mumuni and Fuseini Mumuni. "Data augmentation: A comprehensive survey of modern approaches". In: ***Array*** 16 (2022), p. 100258.

[7] Julius Pfrommer, Thomas Usländer, and Jürgen Beyerer. "KI-Engineering–AI Systems Engineering: Systematic development of AI as part of systems that master complex tasks". In: ***at-Automatisierungstechnik*** 70.9 (2022), pp. 756–766.

[8] Constantin Aliferis and Gyorgy Simon. "Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI". In: ***Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls***. Springer, 2024, pp. 477–524.

[9] Paul Brous, Marijn Janssen, and Riikka Vilminko-Heikkinen. "Coordinating decision-making in data management activities: a systematic review of data governance principles". In: ***Electronic Government: 15th IFIP WG 8.5 International Conference, EGOV 2016, Guimarães, Portugal, September 5-8, 2016, Proceedings 15***. Springer. 2016, pp. 115–125.

[10] Aisha Siddiqa, Ahmad Karim, and Abdullah Gani. "Big data storage technologies: a survey". In: ***Frontiers of Information Technology & Electronic Engineering*** 18 (2017), pp. 1040–1070.

[11] Patrícia Bento, Miguel Neto, and Nadine Côrte-Real. "How data governance frameworks can leverage data-driven decision making: A sustainable approach for data governance in organizations". In: ***2022 17th Iberian Conference on Information Systems and Technologies (CISTI)***. IEEE. 2022, pp. 1–5.

[12] Matthias Schopp. ***Sensorbasierte Zustandsdiagnose und -prognose von Kugelgewindetrieben /***. Forschungsberichte aus dem wbk, Institut für Produktionstechnik, Karlsruher Institut für Technologie (KIT) ; Aachen : Shaker, 2009. URL: http://d-nb.info/999594788/04.

[13] Tobias Schlagenhauf. "Bildbasierte Quantifizierung und Prognose des Verschleißes an Kugelgewindetriebspindeln : Ein Beitrag zur Zustandsüberwachung von Kugelgewindetrieben mittels Methoden des maschinellen Lernens". German. PhD thesis. Karlsruher Institut für Technologie (KIT), 2022. 335 pp. ISBN: 978-3-8440-8875-5. DOI: 10.5445/IR/1000154046.

[14] Tobias Schlagenhauf. ***Ball Screw Drive Surface Defect Dataset for Classification***. 2023. DOI: 10.35097/1511. URL: https://radar.kit.edu/radar/en/dataset/xsvLWXhsaWvzMqkt.

[15] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: ***2017 IEEE International Conference on Computer Vision (ICCV)***. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.


AI
DATA